LIS901a Lecture 3

Thomas Krichel

2000–02–05

Structure

1. The whois++ protocol

Reading

RFC 1835

ROADS web site at http://www.roads.lut.ac.uk/

Protocol and Software

A protocol is a set of ideas on how a device should behave. Such a device may be an individual computer e.g. a web server, or a complete network, say the Internet.

Most protocols used on the Internet are published as Internet RFCs. This stands for Request for Comments. Recently the W3C have also been an important standard setting body.

There is software that implements a protocol. We will use three protocols.

telnet	putty telnet client
ftp	BSD ftp client
http	apache server, netscape client
whois++	ROADS server software

All these are client server protocols. In UNIX the server is implemented in a daemon software.

whois++

Whois++ was initially written to provide a white-page service for the Internet. That idea was before we got junk email.

It was meant to be a extension of the whois program. Nowadays that program delivers domain name and registration information.

In C, when you add write *variable*++, you mean: add 1 to *variable*.

whois++ is a distributed indexing format for attribute/value records. Each attribute value in the whois++ database is divided into one or more words separated by whitespace.

But whois++ does NOT define

- an input syntax for such records
- an attribute or value vocabulary

whois++ allows distributed queries over several servers.

whois++ commands

There are two types of commands

- search commands
  - search terms
  - search constraints
- system commands

whois++ searches

A search command consists of one or more search terms, which might each have local constraints, followed by an optional colon with a set of global search constraints. Each search term operates on every word in the attribute value.

Two or more search terms may be combined with boolean operators AND, OR or NOT (other than the implied AND between terms). The operator AND has higher precedence than the operator OR, but this can be changed by the use of parentheses.

Search constraints that apply to every search term are specified as global constraints. Local constraints override global constraints for the search term they are bound to. The search terms and the global constraints are separated with a colon (':'). Additional global constraints are appended to the end of the search command delimited with a semicolon ';'.

whois++ search term

The general format is any of the following

- A search string. Looks for the string in all values of all records.
- A combination of attribute name, followed by '=', followed by a search string. This is an attribute-value search, that looks for a specific value in a specific attribute only.
- A search term specifier (as on next slide), followed by a '=', followed by a search string

All of these can be followed by an optional semicolon and set of semicolon-separated local constraints.

whois++ search constraints, local (l) or global (g)

name	*functionality*
value	Confine search to attribute values. This is the default.
attribute-value	allows combining attribute and value specifiers in one term.
search-all	Search everything, i.e. attributes and values
handle	Confine search to handles.
template	Confine search to template names.

whois++ search constraints, local (l) or global (g)

name	*value*	
maxhits	1-*max-allowed*	g
search	exact || lsting || regex || fuzzy || substring	lg
case	ignore || consider	lg
format	full || abridged || handle || server-to-ask	g
maxfull	1-*max-allowed*	g
name	*string*	g
language	*As defined in ISO 639:1988*	g
hold		g
ignore	attributelist	g
include	attributelist	g

Sample queries

author=chris and template=user

Find all records where attribute "author" matches "chris" with case ignored. Only USER templates will be searched. An example of a matching record is "Author=Chris Weider".

schoultz and rick;search=lstring

Find all records which have one attribute value matching "schoultz" exactly and one having "rick" as leading substring, both with case ignored. One example is "Name=Rickard Schoultz".

Sample queries

value=phone;search=substring

Find all records which have attribute values matching *phone*, for example the record "Name=Acme telephone inc.", but will not match the attribute name "phone".

search-all=Peter ; search=substring;case=consider

Find all records which have attribute names, template names or attribute values matching "Peter" with respect to case. One example is "Friend-Of-Peter: Yes".

ucdavis;search=substring and (gargano or joan):include=name,email

Find all records which have records containing the words "gargano" or "joan" somewhere in the record, and has the word "ucdavis" somewhere in a word. The result will only show the "name" and "email" fields.

whois++ system commands

commands	List valid whois++ commands supported by this server
constraints	List valid constraints supported by this server
describe	Describe this server, using a standard "Services" template
help	System help, using a "Help" template
list	List templates supported by this system
polled-by	List indexing servers that are know to track this server
polled-for	List information about what this server is tracking for
show *template*	Show contents of *template*
version	return current version of the protocol supported.

server responses

110	Too many hits	The number of matches exceeded the value specified by the maxhits constraint. Server will still reply with as many records as "maxhits" allows.
111	Requested constraint not supported	One or more constraints in query is not implemented. The search is still done.
112	Requested constraint not fullfilled	One or more constraints in query has unacceptable value and was therefore not used. The search is still done.
200	Command Ok	Command accepted and executed. The client must wait for a transaction end message.
201	Command Completed successfully	Command accepted and executed.
203	Bye	Server is closing connection
220	Service Ready	Greeting message. Server is accepting commands.
226	Transaction complete	End of data. All responses to query are sent.
500	Syntax error	
502	Search expression too complicated	This message is sent when the server is not able to resolve a query

polling and forward knowledge

whois++ servers compute a summary of contents called a centroid. The centroid has all attribute names, and all values that there are for the attributes, for each template. Example

Record 1	*Record 2*	*Record 3*
Template: Person	Template: Person	Template: Domain
First-Name: John	First-Name: Joe	Domain-Name: liu.edu
Last-Name: Smith	Last-Name: Smith	Contact-Name: Mike Koenig
Sport: Hockey	Sport: Ice Hockey	

The centroid for this server would be

Record 1	*Record 2*
Template: Person	Template: Domain
First-Name: John Joe	Domain-Name: liu.edu
Last-Name: Smith	Contact-Name: Mike Koenig
Sport: Hockey Ice	